McEliece's Crypto System

C. McFarlane & Y. Sauls

December 8, 2005

Contents

1 Introduction

The goal of the project is to examine a polynomial-time probabilistic algorithm to compute the minimum distance of an arbitrary linear error-correcting code. We will follow the honors thesis of Foster [F] and Irons [I]. Our emphasis will be explaining how to apply this algorithim to decrypt the McEliece cryptosystem. The work shall be done in the GAP coding-theory package GUAVA.

1.1 Background on Cryptosystems

A cryptosystem (or cryptographic system) is the package of all procedures, protocols, cryptographic algorithms and instructions used for encoding and decoding messages using cryptography. Cryptography "the study of the mathematical techniques related to the aspects of information security such as confidentiality, data integrity, entity authenticity, and data origin authentication" [MOV]. Cryptography provides a means for information security and prevents and detects unlawful and malicious activities.

In the 1970s McEliece introduced a cryptosystem based on coding theory. This paper gives an example of this cryptosystem using Gap.

Codewords are the encoded data; they are what is transmitted. A **code** is a set of codewords, and a **linear code** is a subspace of the vector space $(\mathbb{F}_q)^n$, where $\mathbb{F} = GF(q)$ is the finite field with q elements. The **length** of the code is the integer n.

1.2 Error-correcting Codes

Error-correcting codes are used primarily to transmit data across a noisy channel. They do this by encoding the data (adding some redundancy) to the transmission so that the original message can be recovered even if a few errors have occurred.

2 Background

Lemma 1 *Every linear code C of length n and dimension k is permutation equivalent to a code C' with a generator matrix of the form $(I_k \,|\, A)$, where A is a $k \times (n-k)$ matrix. A code with generator matrix in the form above is said to be in standard form.*

Lemma 2

$$d(C) = \min_{\mathbf{c} \in C,\, \mathbf{c} \neq \mathbf{0}} d(\mathbf{0}, \mathbf{c}). \tag{1}$$

Lemma 3 *A linear code C is e-error-correcting if and only if $d(C) \geq 2e+1$. In other words, if $d = d(C)$ then C is $[(d-1)/2]$-error-correcting.*

3 Application to McEliece Public Key Cryptosystem

3.1 Definition of cryptosystem

Alice is sending a message to Bob. Eve is trying to read Alice's message. Given is a $[n, k, d]$ binary code with generator matrix G and also given is a decoding algorithm that corrects up to t errors. Input parameters are an invertible $k \times k$ matrix S and an $n \times n$ permutation matrix P. The triplet (S, G, P) will be the *secret key* and the pair $(t, \hat{G} = SGP)$ will be the *public key.* veryone knows the public key, but Alice and Bob are the only two who know the secret key.

For Alice to transmit a k bit message m, she sends the cipher text $c = m\hat{G} + e$, where e is a random vector of weight at most t. To decipher the cipher text Bob decodes $cP^{-1} = mSG + eP^{-1}$ using the given algorithm.

To decrypt an encoded message, consider the matrix

$$G' = \begin{pmatrix} \hat{G} \\ m\hat{G} + e \end{pmatrix}.$$

This is the generator matrix of a linear code C' with minimum weight codeword e. Note that e is the only vector in C' of minimum weight, by construction. Algorithm 3 in [F] can be modified to yield not only the minimum distance, but also a vector of minimum weight. Using this fact Eve can solve for e, and therefore she can solve for the message m sent by Alice. This example illustrates the usefulness of fast minimum distance algorithms for cracking certain types of McEliece cryptosystems.

3.2 Gap Code: Breaking McEliece

The GAP code below illustrates how to use the `MinimumDistanceRandom` command described in Algorithm 3 in [F] to break the McEliece cryptosystem cipher. We are going to illustrate 3.1 using Gap. We make S and P the identity, and make $\hat{G}$ equal to G. G is the generator matrix of the Reed Solomon code of length 15 over GF(16). c_0 is the random codeword that Alice sent to Bob. To construct the matrix G' in 3.1 we concatenate G with the $w = m\hat{G} + e = c_0 + e$. We denote this matrix by Gnew in Gap, and the code generated by Gnew is called Cnew. Cnew is the C' from section 3.1 We then use the `MinimunDistanceRandomCode` which returns the minimum distance and the minimum weight codeword of the codeword Gnew. We see that the minimium weight codeword of Cnew and e are equal to each other. We can subtract e from w in order to get c_0, which is the codeword that Alice sent to Bob.

```
gap>  F:=GF(16);
GF(2^4)
gap>  C := ReedSolomonCode( 15, 10 );
a cyclic [15,6,10]7..9 Reed-Solomon code over GF(16)
gap>  G:=GeneratorMat(C);;
gap>  g:=Random(SymmetricGroup(15));;
gap>  v:=ShallowCopy(NullWord( 15, F));
[ 0*Z(2), 0*Z(2), 0*Z(2), 0*Z(2), 0*Z(2), 0*Z(2), 0*Z(2), 0*Z(2),
  0*Z(2), 0*Z(2), 0*Z(2), 0*Z(2), 0*Z(2), 0*Z(2), 0*Z(2) ]

gap>  v[1]:=One(F);;
gap>  v[2]:=One(F);;
gap>  v[3]:=One(F);;
gap>  v[4]:=One(F);;
gap>  v[5]:=One(F);;
gap>  e:=Permuted(v,g);;
gap>  c0:=Random(C);;
gap>  w:=VectorCodeword(c0+e);;
gap>  Gnew:=Concatenation(G,[w]);;
gap>  Cnew:=GeneratorMatCode(Gnew,F); time;

a linear [15,7,1..9]6..8 code defined by generator matrix over GF(16)
49109

gap> MinimumDistanceRandom(Cnew,10,13);time;

 This is a probabilistic algorithm which may return the wrong answer.
[ 5, [ 0 0 0 0 1 0 0 1 0 0 1 1 0 0 1 ] ]
158361
gap> e;
[ 0*Z(2), 0*Z(2), 0*Z(2), 0*Z(2), Z(2)^0, 0*Z(2), 0*Z(2), Z(2)^0,
  0*Z(2), 0*Z(2), Z(2)^0, Z(2)^0, 0*Z(2), 0*Z(2), Z(2)^0 ]
```

This is a linear [15,7] code defined by generator matrix over GF(16)

```
gap> MinimumDistanceRandom(Cnew,10,13);time;

 This is a probabilistic algorithm which may return the wrong answer.

[ 5, [ 0 0 0 0 1 0 0 1 0 0 1 1 0 0 1 ] ]

158361
```

The second component returned is a minimum weight vector of the code Cnew. This is the vector that we need to crack the cipher. Notice the error vector **e** matches the minimum weight vector returned by `MinimumDistanceRandom`, as desired.

References

[F] A. Foster, "A polynomial-time probabilistic algorithm for the minimum distance of an arbitrary linear error-correcting code," Mathematics Honors Report, Spring 2003-2004.

[GAP] GAP 4.4, Groups Algorithms Programming, version 4.4 `http://www.gap-system.org`

[GUA] GUAVA, `http://cadigweb.ew.usna.edu/~wdj/gap/GUAVA/`

[I] W. Irons, "A polynomial-time probabilistic algorithm for the minimum distance of a non-binary linear error-correcting code" Mathematics Honors Report, Spring 2004-2005.

[M] R. J. McEliece, "A public-key cryptosystem based on algebraic coding theory", vol. 42-44, DSN Progress Rep., 1978.

[MOV] A. Meneezes, P. van Oorschot, and S. Vanstone, **Handbook of Applied Crytpography**. CRC Press, 1996.